

# ***F<sub>i</sub>NK*** – the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> File Name Keeper\*

Didier Verna

<mailto:didier@lrde.epita.fr>

<http://www.lrde.epita.fr/~didier/>

November 14, 2007

## 1 Description

This package is a real fink indeed: it looks over your shoulder and keeps track of files `\input`'ed (the L<sup>A</sup>T<sub>E</sub>X way) or `\include`'ed in your document. You then have a permanent access to the directory, name and extension of the file currently being processed through several macros. Dis packache fas orichinally a hack dat I used somefere elss, but since it might be off a cheneral interest, I'fe decided to make it a separate fink...

The ***F<sub>i</sub>NK*** package is Copyright © 1999, 2000, 2001, 2002, 2007 Didier Verna, and distributed under the terms of the LPPL license.

## 2 User Interface

To use the package, simply say `\usepackage[options]{fink}` in the preamble of your document. This will do everything for you. Available options will be described when appropriate.

### 2.1 Retrieving the current file's name components

<code>\finkdir</code>	The file currently being processed is described by the macros <code>\finkdir</code> , <code>\finkbase</code> and <code>\finkext</code> which expand (as you may have guessed) to the directory, base name (sans extension), and extension of the file.
<code>\finkbase</code>	
<code>\finkext</code>	
<code>\finkfile</code>	Additionally, the macro <code>\finkfile</code> is defined to be <code>\finkbase.\finkext</code> (as in previous versions), and the macro <code>\finkpath</code> (new in version 2.0) is defined to be <code>\finkdir\finkfile</code> . Feel free to use these macros in your sources.
<code>\finkpath</code>	

### 2.2 Main file's name components

<code>maindir</code>	Because there's no way T <sub>E</sub> X can give you back information about the file being processed (apart from its base name), <i><b>F<sub>i</sub>NK</b></i> provides the options <code>maindir</code> (defaults to <code>./</code> ) and <code>mainext</code> (defaults to <code>tex</code> ) for changing the directory and the extension
<code>mainext</code>	

---

\*This document describes ***F<sub>i</sub>NK*** 2.1, release date 2007/11/14.

of the main source file. For instance, suppose your source file is in `src/foo.ltx` and you are compiling in `pdf/`. You can then use the package as follows:

```
\usepackage[maindir=../src,mainext=ltx]{fink}
```

### 3 AUC- $\text{\TeX}$ support

AUC- $\text{\TeX}$  is a powerful major mode for editing  $\text{\TeX}$  documents in Emacs or XEmacs. In particular, it provides automatic completion of macro names once they are known. *F<sub>i</sub>NK* supports AUC- $\text{\TeX}$  by providing a style file named `fink.el` which contains AUC- $\text{\TeX}$  definitions for the relevant macros. This file should be installed to a location where AUC- $\text{\TeX}$  can find it (usually in a subdirectory of your  $\text{\LaTeX}$  styles directory). Please refer to the AUC- $\text{\TeX}$  documentation for more information on this.

### 4 Caveat

*F<sub>i</sub>NK* cannot follow files included with the  $\text{\TeX}$  `\input` primitive. That's because  $\text{\TeX}$  has a very insensible way of defining primitives whose argument parsing syntax is not available for macros. As a consequence, it's almost impossible to redefine the `\input` primitive without breaking its syntax (one would have to parse the characters one by one, and I'm not ready to do so...). *F<sub>i</sub>NK* currently does not follow auxiliary files either.

### 5 Changes

- v2.1 Fix bug preventing expansion in math mode, reported by Alain Schremmer, fixed by Morten Hoegholm before I could even raise my little finger.
- v2.0 New macros `\finkdir`, `\finkbase`, `\finkext` and `\finkpath` suggested by Alain Schremmer  
New options `mainext` and `maindir`, use `kvoptions` for options management
- v1.2 Fixed conflict with `\includegraphics`, reported by Jim Crumley
- v1.1 Fixed missing 3rd arg to `\PackageError` call from `\finkextension`

### 6 The Code

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{fink}[2007/11/14 v2.1
3                               Keep track of the current filename]
4
5 \RequirePackage{kvoptions}
6 \SetupKeyvalOptions{family=fnk,prefix=fnk@}
7
```

#### 6.1 Main file initial settings

```
maindir
mainext
```

```

8 \DeclareStringOption[\@currdir]{maindir}
9 \DeclareStringOption[tex]{mainext}
10
The following is for backward compatibility only (not documented anymore). It
provides support for the old tex and ltx options (still fonctionnal), and for the
\finkextension macro. However, this macro is now defined to trigger an error,
begging the user to use the new option instead.
11 \newcommand*\@fink@mainext[1]{\setkeys{fnk}{mainext={#1}}}
12 \newcommand*\fink@mainext{%
13   \expandafter\@fink@mainext\expandafter{\CurrentOption}}
14 \DeclareVoidOption{tex}{\fink@mainext}
15 \DeclareVoidOption{ltx}{\fink@mainext}
16
17 \newcommand*\finkextension[1]{%
18   \PackageError{FiNK}{%
19     \protect\finkextension\space shouldn't be used anymore.\MessageBreak
20     Please use the 'mainext' package option instead.}%
21     No big deal right ?\MessageBreak
22     Type X to quit and modify your source.}}
23 \@onlypreamble\finkextension
24
25 \ProcessKeyvalOptions*
26

```

## 6.2 File's name components macros

```

\finkdir We declare the user-level macros here. \fink@file is used to compute file names,
\finkbase possibly with no extension.
\finkext
\finkfile 27 \newcommand*\finkdir{\fnk@maindir}
\finkpath 28 \newcommand*\finkbase{\jobname}
29 \newcommand*\finkext{\fnk@mainext}
30
31 \newcommand*\finkfile{
32 \newcommand*\fink@file[2]{#1\ifx\#2\\\else.#2\fi}
33 \xdef\finkfile{\fink@file{\jobname}{\fnk@mainext}}
34
35 \newcommand*\finkpath{
36 \xdef\finkpath{\finkdir\finkfile}
37
38 \PackageInfo{FiNK}{main file set to "\finkpath"}
39

```

## 6.3 Commands overriding

`\fink@prepare` This macro prepares the name of next file to be input. We arrange to setup a complete filename, including directory and extension.

As of version 1.2, this macro performs in a group of its own. This fixes a problem that appeared when using `\includegraphics` with a filename with an explicit extension. `\includegraphics` calls `\filename@parse` itself, so it is important that the same call in `\fink@prepare` only have a local effect, just the time to compute the new values for the `\fink@next*` macros.

```

40 \newcommand*\fink@prepare[1]{%
41   \begingroup%
42     \filename@parse{#1}%
43   \xdef\fink@nextdir{%
44     \ifx\filename@area\@empty%
45       \fnk@maindir%
46     \else%
47       \fnk@maindir\filename@area%
48     \fi}%
49   \xdef\fink@nextbase{\filename@base}%
50   \xdef\fink@nexttext{\ifx\filename@ext\relax tex\else\filename@ext\fi}%
51   \xdef\fink@nextfile{\fink@file{\fink@nextbase}{\fink@nexttext}}%
52   \xdef\fink@nextpath{\fink@nextdir\fink@nextfile}%
53   \endgroup}
54

```

`\fink@input` These macros are defined for a convenient use of `\expandafter`. They save and  
`\fink@restore` restore the current filename. Remember that `\@@input` is L<sup>A</sup>T<sub>E</sub>X's redefinition of  
the T<sub>E</sub>X input primitive.

```

55 \newcommand*\fink@input{%
56   \xdef\finkdir{\fink@nextdir}%
57   \xdef\finkbase{\fink@nextbase}%
58   \xdef\finkext{\fink@nexttext}%
59   \xdef\finkfile{\fink@nextfile}%
60   \xdef\finkpath{\fink@nextpath}%
61   \@@input\@filef@und}
62 \newcommand*\fink@restore[1]{%
63   \begingroup%
64     \filename@parse{#1}%
65   \xdef\finkdir{\filename@area}%
66   \xdef\finkbase{\filename@base}%
67   \xdef\finkext{\filename@ext}%
68   \xdef\finkfile{\fink@file{\finkbase}{\finkext}}
69   \xdef\finkpath{\finkdir\finkfile}%
70   \endgroup}
71

```

Note: in earlier versions, we redefined `\IfFileExists` to prepare the name of the next file, but this is bad because it can be used outside of *F<sub>i</sub>NK*'s scope. We also redefined `\@input`, but neither `\include` nor `\input` use it.

`\InputIfFileExists` L<sup>A</sup>T<sub>E</sub>X's `\input` and `\include` commands use `\InputIfFileExists`, so let's re-define it here:

```

72 \long\def\InputIfFileExists#1#2{%
73   \IfFileExists{#1}{%
74     #2\@addtofilelist{#1}%
75     \fink@prepare{#1}%
76     \expandafter\fink@input%
77     \expandafter\fink@restore\expandafter{\finkpath}}%
78

```

Well, I think that's it. Enjoy using *F<sub>i</sub>NK*!

Copyright © 1999, 2000, 2001, 2002, 2007 Didier Verna.